CLAIMS

What is claimed is:

1.   A method of accessing a particular entry in a list of entries in a computer system, each of the entries including a next entry pointer that points to another entry in the list such that the

5   next entry pointers together form a closed loop, the method comprising:

reading a start pointer of one of the entries;

examining the entries in the list in turn beginning with the entry pointed to by the start pointer and continuing until the particular entry is found;

accessing the particular entity; and

10   overwriting the start pointer so as to point to the particular entry.

2.   A method as claimed in claim 1, further comprising terminating examination of the entries in the event that the particular entry is not found and all of the entries in the list have been examined.

3.   A method as claimed in claim 1, wherein accessing the particular entry includes

15   reading the entry.

4.   A method as claimed in claim 1, wherein accessing the particular entry includes writing to the entry and the method further comprises imposing a mutex

5.   A method as claimed in claim 1, wherein overwriting the start pointer does not require a mutex.

20   6.   A method as claimed in claim 1, wherein the computer system is a multi-threaded environment and the overwriting step is performed atomically.

7.   A method of accessing a particular entry in a list of entries in a computer system, each of the entries including a next entry pointer that points to another entry in the list such that the next entry pointers together form a closed loop, the method comprising:

25   reading a start pointer of one of the entries;

examining the entries in the list in turn beginning with the entry pointed to by the start pointer and continuing until the particular entry is found;

if the particular entry is found then performing the following steps:

accessing the particular entity;

overwriting the start pointer so as to point to the particular entry; and

if the particular entry is not found and all of the entries in the list have been examined then performing the following step:

terminating examination of the entries.

5   8.   A method as claimed in claim 7, wherein accessing the particular entry includes reading the entry.

9.   A method as claimed in claim 7, wherein accessing the particular entry includes writing to the entry and the method further comprises imposing a mutex

10.   A method as claimed in claim 7, wherein overwriting the start pointer does not require
10   a mutex.

11.   A method as claimed in claim 7, wherein the computer system is a multi-threaded environment and the overwriting step is performed atomically.

12.   A method of accessing a particular entry in a list of entries in a computer system, each of the entries including a next entry reference that refers to another entry in the list such that
15   the next entry references together form a closed loop, the method comprising:

reading a start reference of one of the entries;

examining the entries in the list in turn beginning with the entry pointed to by the start reference and continuing until the particular entry is found;

accessing the particular entity; and

20   overwriting the start reference so as to point to the particular entry;

Sub A 10.   A method as claimed in claim 9, further comprising terminating examination of the entries in the event that the particular entry is not found and all of the entries in the list have been examined.

11.   A method as claimed in claim 9, wherein the accessing and examining steps are
25   omitted when the particular entry is not found.

12.   A method as claimed in claim 9, wherein accessing the particular entry includes reading the entry.

13.   A method as claimed in claim 9, wherein accessing the particular entry includes writing to the entry and the method further comprises imposing a mutex

14. A method as claimed in claim 9, wherein overwriting the start pointer does not require a mutex.

15. A method as claimed in claim 9, wherein the computer system is a multi-threaded environment and the overwriting step is performed atomically.

16. A storage structure comprising

a list of entries, each entry having a next entry pointer and each next entry pointer pointing to another entry such that the next entry pointers together form a closed loop; and

an overwritable start pointer that points to an entry in the list of entries, the start pointer capable of being overwritten to point to an entry other than the entry.

17. A storage structure as claimed in claim 16, wherein the entry other than the entry is a particular entry that was searched for and found.

18. A storage structure as claimed in claim 17, wherein the particular entry is searched for by examining the list of entries in turn beginning with an entry pointed to by the start pointer and continuing until the particular entry is found.